RMDIR

Vulnerable to TOCTOU issues

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-04

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6953 bytes

Attack Category	• Path spoofing or confusion problem
Vulnerability Category	• Indeterminate File/Path • TOCTOU - Time of Check, Time of Use
Software Context	• File Management
Location	
Description	The rmdir function attempts to remove a directory. It is generally vulnerable to classic TOCTOU attacks. A call to rmdir() should be flagged if the first argument (the directory) is used earlier in a check-category call.
APIs	**Function Name** / **Comments** _rmdir / use; win32 _trmdir / use; win32 _wrmdir / use; win32 rmdir / use rmdirp / use; Solaris
Method of Attack	The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results. The mkdir() call is a use-category call, which when preceded by a check-category call can be indicative of a TOCTOU vulnerability. A TOCTOU attack in regards to rmdir() can occur when a check for the existence of a directory occurs and then a directory is deleted. Between the two actions, an attacker could, for example, link the

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	target directory (the one to be deleted) to a known directory. The subsequent deletion of the target directory would attempt to delete the attacked directory.
Exception Criteria	
Solutions	

Solution Applicability	Solution Description	Solution Efficacy
Generally applicable to any rmdir.	Utilize a file descriptor version of stat/ fstat when checking.	Effective.
Generally applicable to any rmdir.	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check. Don't perform a check prior to use. Attempt to remove the directory and then check status after the creation.	Does not resolve the underlying vulnerability but limits the false sense of security given by the check. Checking the status after the operation does not change the fact that the operation may have been exploited but it does allow halting of the application in an error state to help limit further damage.
Generally applicable to any rmdir.	Limit the interleaving of operations on files from multiple processes.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.

	Generally applicable to any rmdir.	Limit the spread of time (cycles) between the check and use of a resource.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Generally applicable to any rmdir.	Recheck the resource after the use call to verify that the action was taken appropriately.	Effective in some cases.
Signature Details	int rmdir(const char *path);		

Examples of Incorrect Code	`#include <sys/types.h>`
	`#include <sys/stat.h>`
	
	`int check_status;`
	`int use_status;`
	`struct stat statbuf;`
	`...`
	
	`check_status=stat("tobedeleteddir", &statbuf);`
	
	`...`
	`<long enough intervening code>`
	
	`use_status=rmdir("tobedeleteddir");`		

Examples of Corrected Code	`// solution to thread exclusion version.`
	`// TOCTOU attacks in general are very broad, some solutions are probabilistic and this likely should be`
	`// considered for a set of rules and fixes instead of one rule.`
	
	`pthread_mutex_lock(&mutex);`
	`//lock`
	
	`check_status=stat("tobedeleteddir", &statbuf);`
	`/...`
	`use_status=rmdir("tobedeleteddir");`
	
	`//unlock`
	`pthread_mutex_unlock(&mutex);`		

Source References	• Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, ch. 9.		

	• man page for rmdir()	
Recommended Resource		
Discriminant Set	**Operating Systems**	• UNIX • Windows
	Language	

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com